THE UNIVERSITY *of* ADELAIDE

Data Quality for ML Based Software Security Solutions: Lessons and Recommendations

Ali Babar

CREST – Centre for Research on Engineering Software Technologies

Auburn University, 5th October, 2023

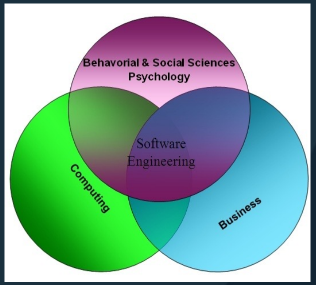
Brief Bio

M. Ali Babar

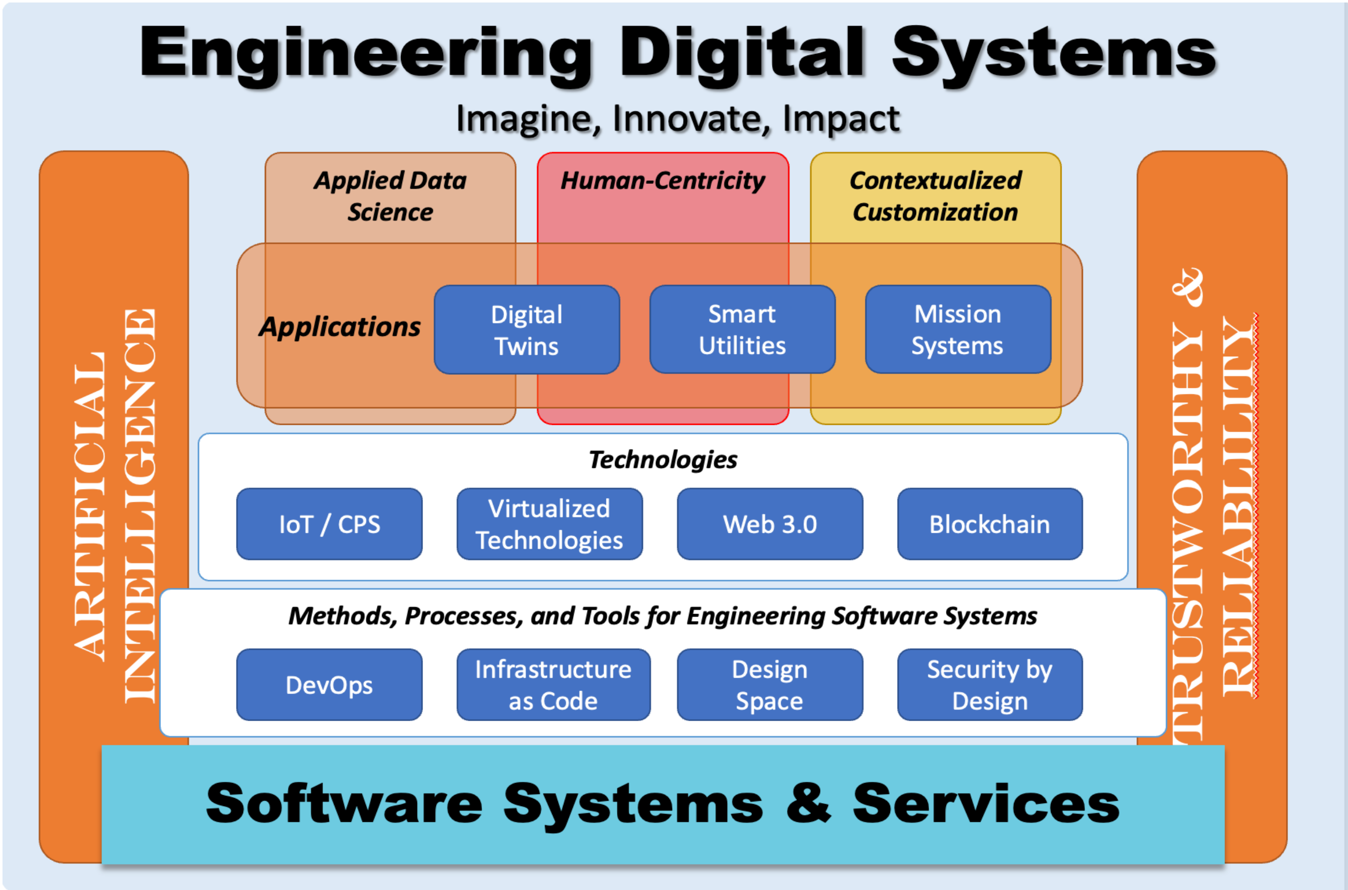
- Professor, School of Computer Science, University of Adelaide, Australia – Nov. 2013 -
- Founding Lead – The Centre for Research on Software Technologies (CREST) – Nov 2013 –
- **Theme Lead – Platforms and Architectures for Security as Service, Cyber Security Cooperative Research Centre (CSCRC)**
- For current research areas: please visit CREST website: crest-centre.net

Previous Work History

- Senior research and academic positions in UK, Denmark & Ireland



THE UNIVERSITY *of* ADELAIDE



THE UNIVERSITY of ADELAIDE

Talk's Roadmap

- CREST's Data Centric Software Security Research
- Data-Centric Software Security Quality Assurance
- Data Quality Problems Experienced/Observed
- Some Recommendations to Deal with Challenges

Software/AI is Everywhere

Pay

Uber
Eats

CBS NEWS NEWS SHOWS LIVE LOCAL Login

POLITICS ›

Nightmare before Christmas: What to know about the Log4j vulnerability

National Cyber Security Centre

ABOUT NCSC CISP REPORT

Home Information for... Advice & guidance Education & skills Products & services News

Home

NEWS

Alert: Apache Log4j vulnerabilities

The NCSC is advising organisations to take steps to mitigate the Apache Log4j vulnerabilities.

Australians warned of widespread Log4j software vulnerability

'The exact extent of the exposure is still unravelling.'

9NEWS

National Latest Politics World Videos Live Today Show ACA

News / Technology

What you need to know about the security flaw that could impact the entire internet

THE UNIVERSITY of ADELAIDE

Software Vulnerabilities and Cybersecurity Incidents

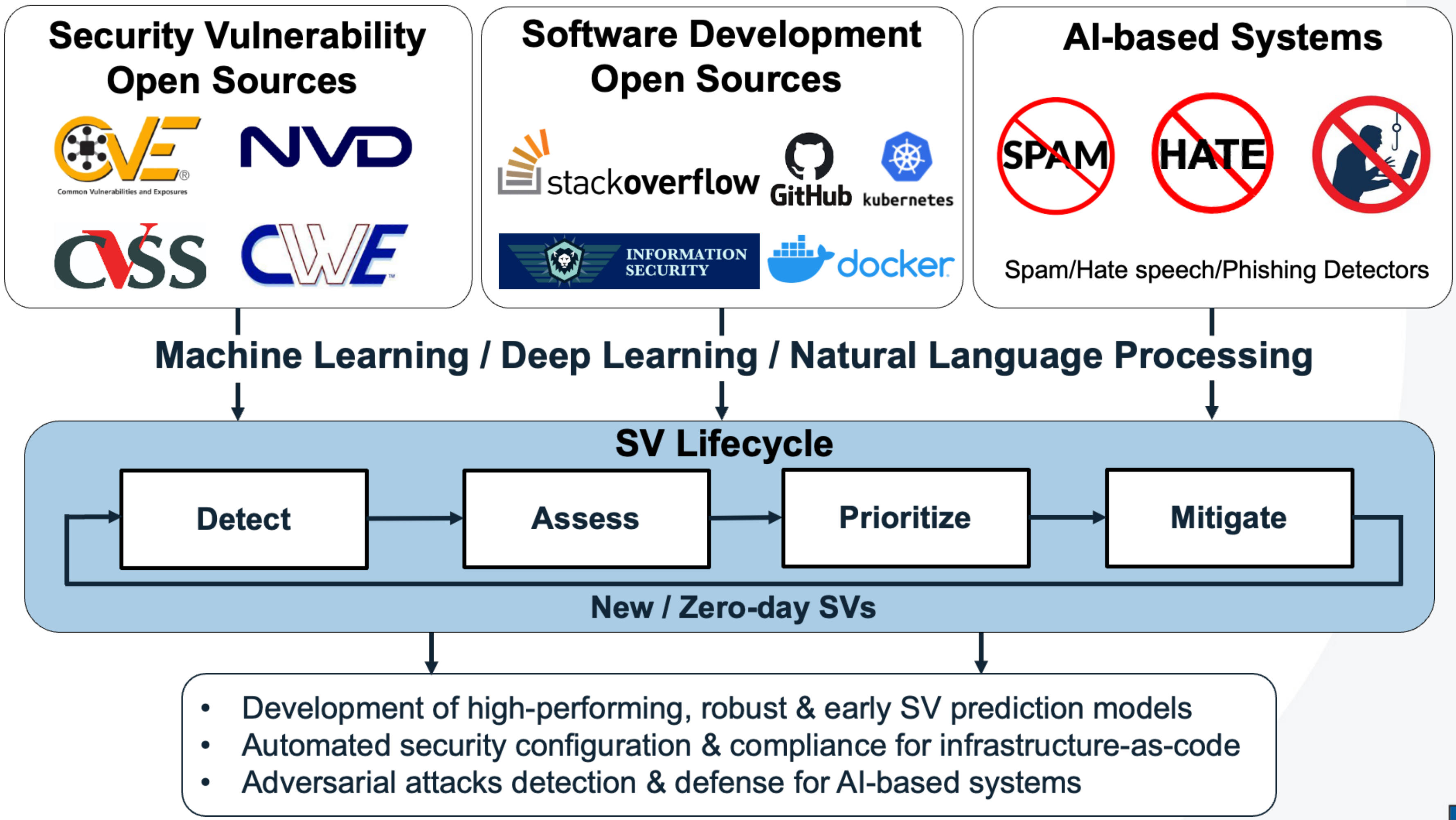
- Approximately 90% of cyber incidents are caused by the vulnerabilities rooted in software – proprietary or sourced

- Software Bill Of Material (SBOM) is becoming ineffective in answering critical questions

  - Q1: Do we really know what's in software coming into the organisation?

  - Q2: How do we establish trust and preserve the security of software coming into the organisation?

THE UNIVERSITY of ADELAIDE

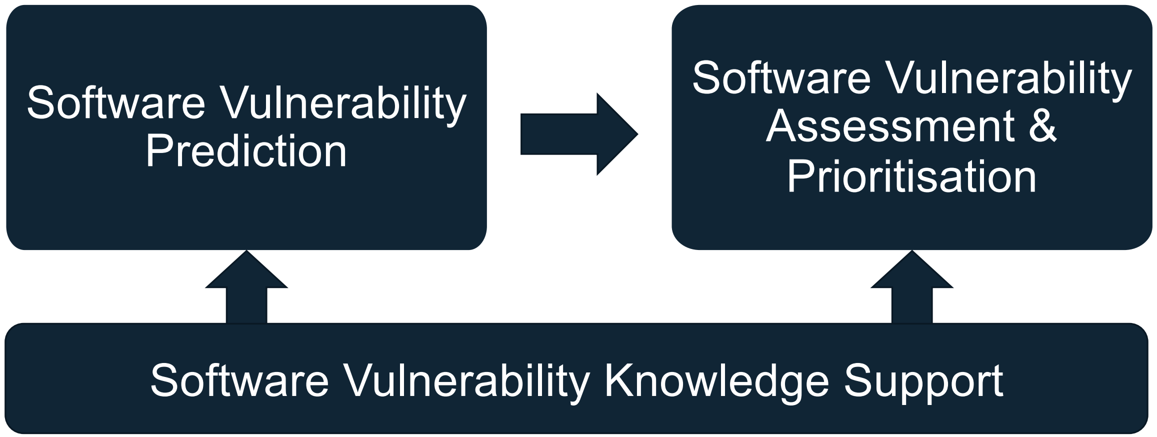


THE UNIVERSITY of ADELAIDE

Data-Driven Software Security at CREST

1. LineVD: statement-level vulnerability detection using graph neural networks (MSR '22)
2. Noisy label learning for security defects (MSR '22)
3. KGSecConfig: A Knowledge Graph Based Approach for Secured Container Orchestrator Configuration (SANER '22)
4. An empirical study of rule-based and learning-based approaches for static application security testing (ESEM '21)

1. A survey on data-driven software vulnerability assessment and prioritization (CSUR '22)
2. On the use of fine-grained vulnerable code statements for software vulnerability assessment models (MSR '22)
3. An investigation into inconsistency of software vulnerability severity across data sources (SANER '22)
4. DeepCVA: Automated commit-level vulnerability assessment with deep multi-task learning (ASE '21)
5. Automated software vulnerability assessment with concept drift (MSR '19)

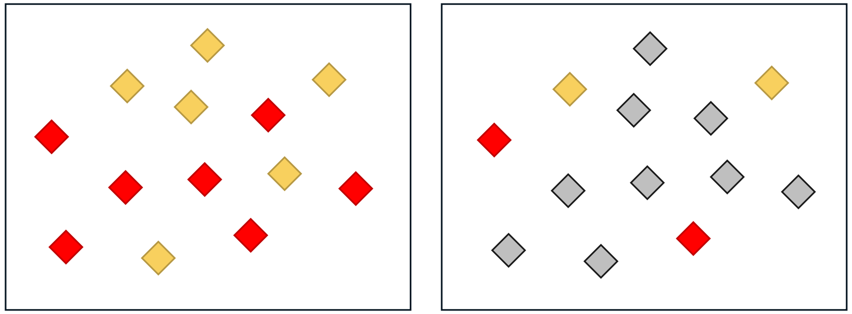


1. An empirical study of developers' discussions about security challenges of different programming languages (EMSE '22)
2. Well begun is half done: an empirical study of exploitability & impact of base-image vulnerabilities (SANER '22)
3. PUMiner: Mining security posts from developer question and answer websites with PU learning (MSR '20)
4. A large-scale study of security vulnerability support on developer Q&A websites (EASE '21)

THE UNIVERSITY of ADELAIDE

Data Quality for Data-Driven Software Security

Assumption **Reality**



No perfectly clean dataset of vulnerabilities:

- **Label noise**
  - False positives of data collection
  - Constantly discovered new vulnerabilities
- **Data noise (e.g., duplicates)**

Data Quality
Assessment & Analysis

GOALS

Robust & noise-tolerant
learning techniques

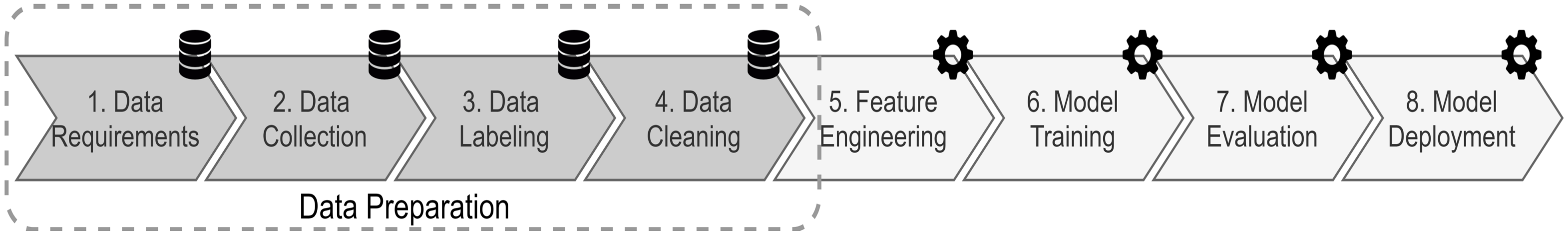
1. Data Quality for Software Vulnerability Datasets, ICSE '23 (CORE A*)
2. Data preparation for software vulnerability prediction: A systematic literature review, TSE '22 (A*)
3. Noisy label learning for security defects, MSR '22 (CORE A)
4. An investigation into inconsistency of software vulnerability severity across data sources, SANER '22 (CORE A)

CYBER SECURITY COOPERATIVE RESEARCH CENTRE
CREST
THE UNIVERSITY of ADELAIDE

Data-Centric Software Security Assurance

THE UNIVERSITY *of* ADELAIDE

Data Preparation for ML Based Security Solutions



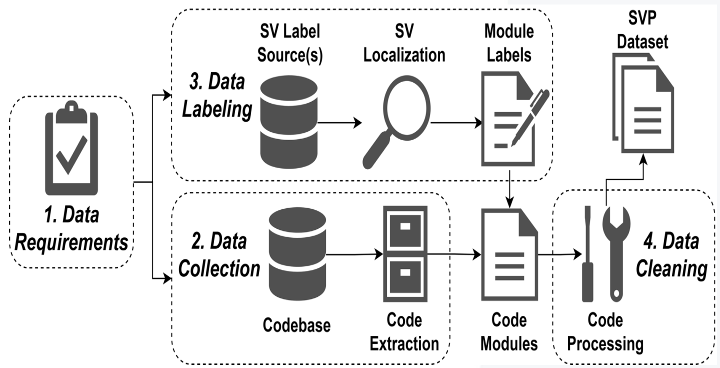
- Data Requirements determine the types and source of data for building a model
- “Data Wrangling” (collection, labelling and cleaning) steps of ML Workflow
- “Data Wrangling” (or preparation) can take up to 25% of an industry project time

1. Data Quality for Software Vulnerability Datasets, ICSE '23 (CORE A*)
2. Data preparation for software vulnerability prediction: A systematic literature review, TSE '22 (A*)
3. Noisy label learning for security defects, MSR '22 (CORE A)
4. An investigation into inconsistency of software vulnerability severity across data sources, SANER '22 (CORE A)

THE UNIVERSITY of ADELAIDE

Data-Centric Software Security Assurance

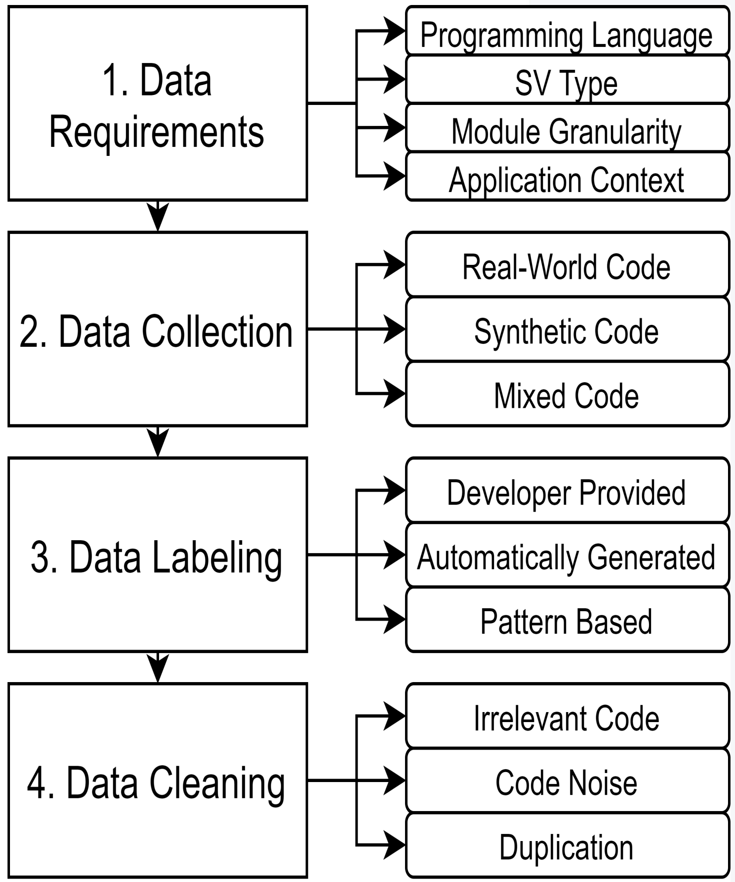
- Software Vulnerabilities Prediction (SVP) approaches purport to learn from history and predict SV
- Prediction approaches are becoming popular as early lifecycle software security assurance techniques
- SVP models may or may not analyse program syntax and semantic; the latter leverages DL
- Being ML dependent, SVP needs data preparation as per the workflow of ML shown on the last slide
- SVP data preparation needs several important consideration including source and labelling



1. Data Quality for Software Vulnerability Datasets, ICSE '23 (CORE A*)
2. Data preparation for software vulnerability prediction: A systematic literature review, TSE '22 (A*)
3. Noisy label learning for security defects, MSR '22 (CORE A)
4. An investigation into inconsistency of software vulnerability severity across data sources, SANER '22 (CORE A)

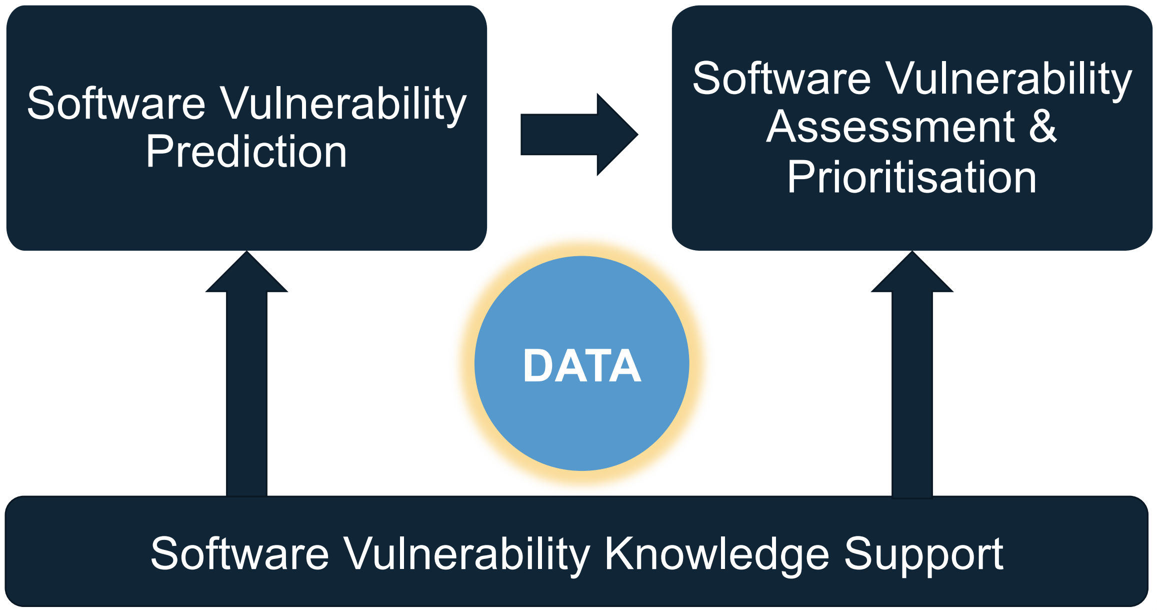
Data Preparation Consideration for SVP

- Data requirements vary depending upon the context and capabilities needed of a ML model
- Data may be collected from real-world, synthetic code or mixed code – training/testing model
  - Trade-off between scarcity and realism
- Gathered data need labelling – provided by developers (NVD), tools based, or based on patterns
  - Labelling non-vulnerable class is problematic
- Data cleaning is required for a certain format and reducing noise from collected/labelled data



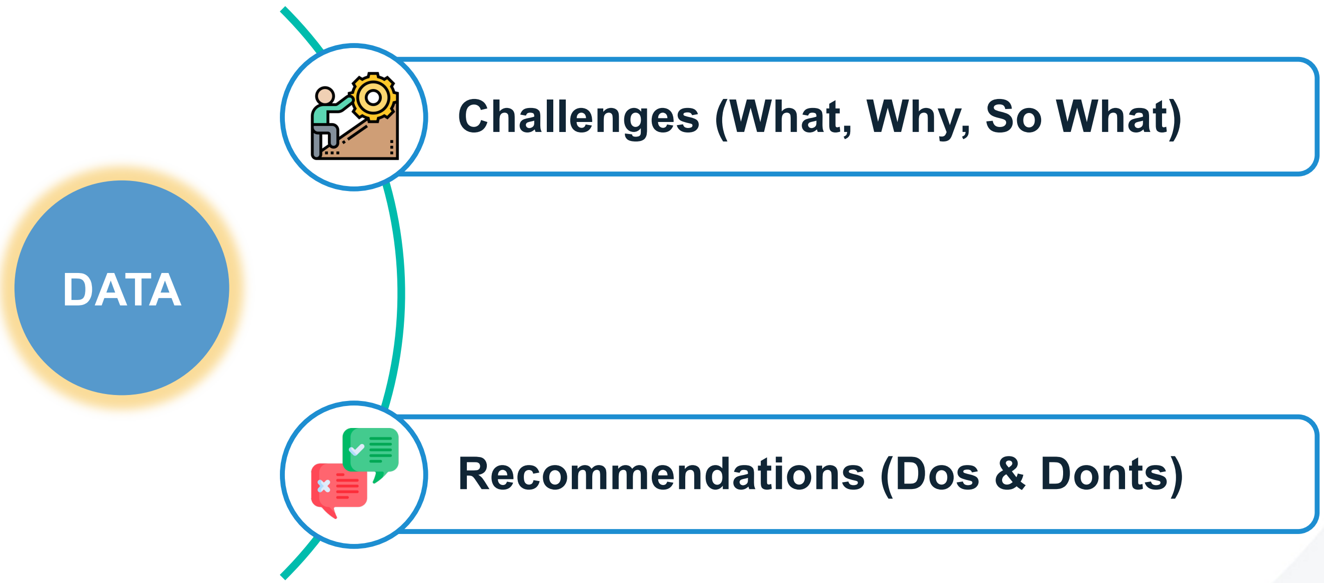
Data Quality Challenges in MLC

Data-Driven Software Security at CREST



THE UNIVERSITY of ADELAIDE

Data Quality for Data-Driven Software Security



THE UNIVERSITY *of* ADELAIDE

Security Data Challenges

- Scarcity
- (In)Accuracy
- (Ir)Relevance
- Redundancy
- (Mis)Coverage
- Non-Representative-ness
- Drift
- (In)Accessibility
- (Re)Use
- Maliciousness

Security Data Challenges

Scarcity

(In)Accuracy

(Ir)Relevance

Redundancy

(Mis)Coverage

Non-Representative-ness

Drift

(In)Accessibility

(Re)Use

Maliciousness

Info (What)

Cause (Why)

Impact (So What)

Data Scarcity

Info

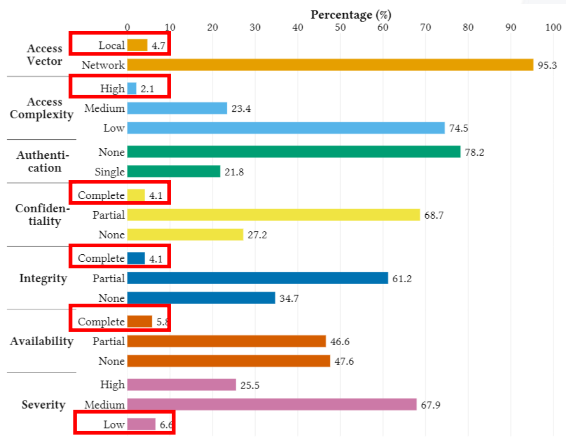
- *Hundreds/Thousands* security issues vs. *Million* images
- Security issues < 10% of all reported issues (even worse for new projects)

Causes

- Lack of explicit labeling/understanding of security issues
- Imperfect data collection (precision vs. recall vs. effort)
- Rare occurence of certain types of security issues

Impacts

- Leading to imbalanced data
- Lacking data to train high-performing ML models
- *More data beats a cleverer algorithm*



R. Croft, et al., *Proceedings of the IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, 2022, 435-447.
T. H. M. Le, et al., *Proceedings of the 17th International Conference on Mining Software Repositories (MSR)*, 2020, 350-361.

THE UNIVERSITY *of* ADELAIDE

Data (In)Accuracy

Info

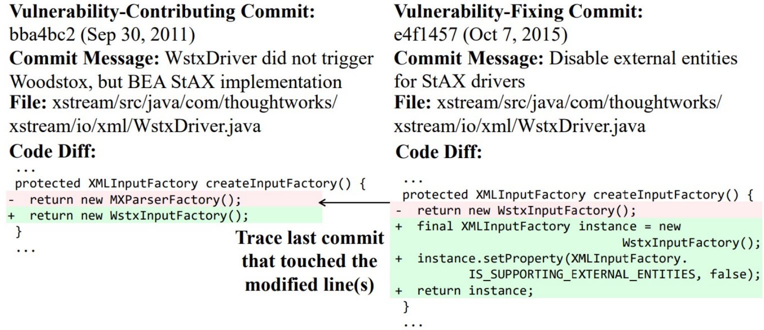
- (Non-)Security issues *not* labelled as such
- FN: Critical vulnerabilities *unfixed* for long time (> 3 yrs)
- FP: Wasting inspection effort

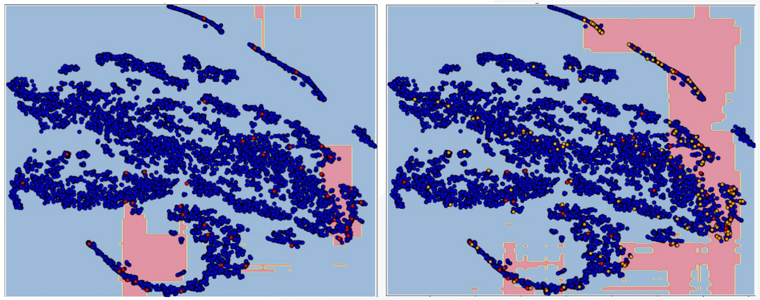
Causes

- *We don't know what we don't know (unknown unknowns)*
- Lack of reporting or silent patch of security issues
- Tangled changes (fixing non-sec. & sec. issues together)

Impacts

- Criticality: Systematic labeling errors > random errors
- Making ML models learn the wrong patterns
- Introducing backdoors of ML models





Without latent SVs | With latent SVs

THE UNIVERSITY of ADELAIDE

R. Croft et al., *Proceedings of the IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, 121-133.
R. Croft, et al., *Proceedings of the IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, 2022, 435-447.
T. H. M. Le, et al., *Proceedings of the 36th International Conference on Automated Software Engineering (ASE)*, 2021, 717-729.

Data (Ir)Relevance

Info

- Not all input is useful for predicting security issues
- Ex1: Code comments for predicting vulnerabilities?!
- Ex2: A file containing fixed version update is vulnerable?!

Causes

- Lack of data exploratory analysis
- Lack of domain expertise (e.g., NLPers working in SSE)
- Trying to beat that state-of-the-art

Impacts

- Negatively affecting the construct validity
- Reducing model performance (e.g., code comments reduced SVP performance by 7x in Python)

Security fixes (87c89f0) in the Apache jspwiki project

THE UNIVERSITY of ADELAIDE

T. H. M. Le, et al., *Proceedings of the IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, 2022, 621-633.
T. H. M. Le, et al., *Proceedings of the 36th International Conference on Automated Software Engineering (ASE)*, 2021, 717-729.

Data Redundancy

Info

- Same security issues found across different software versions, branches, & even projects

Causes

- Cloned projects from mature projects (e.g., Linux kernel)
- Merged code from feature branches into the master branch
- Renamed files/functions with the same code content
- Cosmetic-only changes (different white spaces or new lines)

Impacts

- Limiting learning capabilities of ML models
- Leading to bias and overfitting for ML models
- Inflating model performance (same training/testing samples)

Thousands of cloned projects sharing same vulns as the Linux kernel

Dataset	With Duplication	Without duplication	Change
Big-Vul	0.826	0.816	1.2% ↓
Devign	0.285	0.247	13.3% ↓
D2A	0.748	0.020	97.3% ↓
Juliet	0.910	0.861	5.5% ↓

Vulnerability prediction performance *before* & *after* removing the redundancies in common datasets

THE UNIVERSITY *of* ADELAIDE

R. Croft et al., *Proceedings of the IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, 121-133.
T. H. M. Le, et al., *Proceedings of the IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, 2022, 621-633.
R. Croft, et al., *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2021, 1-12.

Data (Mis)Coverage

Info

- Security issues spanning multiple lines, functions, files, modules, versions, & even projects, but...
- Current approaches mostly focus on single function/file

Causes

- Partial security fixes in multiple versions
- For ML: coverage vs. size (↑ as granularity ↓)
- For Devs: coverage vs. convenience (inspection effort)
- Fixed-size (truncated) input required for some (DL) models

Impacts

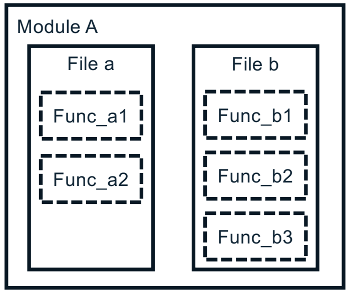
- Lacking context for training ML models to detect complex (e.g., intra-function/file/module) security issues
- Incomplete data for ML as security-related info. truncated

User's malicious input? How to know using only the current function?

```
protected Object getOverrideExpr(ActionInvocation invocation, Object value) {
    return "'" + value + "'";
    return escape(value);
}
```

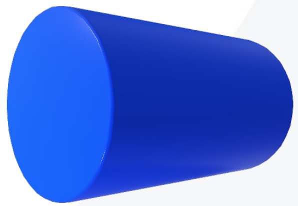
Deleted line

Added line



Assume that Module A is vulnerable then:

- Vuln. module: 1
- Vuln. files: 2
- Vuln. functions: 5



Cylinder vs. Circle vs. Rectangle?

THE UNIVERSITY of ADELAIDE

T. H. M. Le, et al., *Proceedings of the IEEE/ACM 19th International Conference on Mining Software Repositories (MSR)*, 2022, 621-633.
T. H. M. Le, et al., *Proceedings of the 36th International Conference on Automated Software Engineering (ASE)*, 2021, 717-729.

Data (Non-)Representativeness

Info

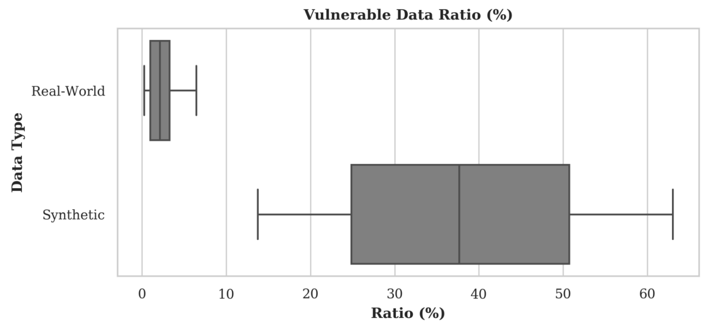
- Real-world security issues (e.g., NVD) vastly different from synthetic ones (e.g., SARD)
- Varying characteristics of security issues across projects

Causes

- Synthetic data: local & created by pre-defined rules
- Real-world data: inter-dependent & complex
- Different features & nature between apps

Impacts

- Perf. (F1) gap: Synthetic (0.85) vs. Real-world (0.15)
- Lack of generalisability & transferability of ML models
- Within-project prediction > Cross-project prediction



Synthetic data

Technique	Training	Acc	Prec	Recall	F1
VulDeePecker	NVD/SARD	-	86.90	-	85.40
SySeVR	NVD/SARD	95.90	82.50	-	85.20
Russell *et al.*	Juliet	-	-	-	84.00
	Draper	-	-	-	56.6
Devign	FFMPeg+Qemu	72.26	-	-	73.26

Real-world data

Approach	Acc	Prec	Recall	F1
Russell *et al.*	90.98 *(0.75)*	24.63 *(5.35)*	10.91 *(2.47)*	15.24 *(2.74)*
VulDeePecker	89.05 *(0.80)*	17.68 *(7.51)*	13.87 *(8.53)*	15.7 *(6.41)*
SySeVR	84.22 *(2.48)*	24.46 *(4.85)*	40.11 *(4.71)*	30.25 *(2.35)*
Devign	88.41 *(0.66)*	34.61 *(3.24)*	26.67 *(6.01)*	29.87 *(4.34)*

R. Croft, et al., *IEEE Transactions on Software Engineering*, 2022.
S. Chakraborty, et al., *IEEE Transactions on Software Engineering*, 2021.

THE UNIVERSITY *of* ADELAIDE

Data Drift

Info

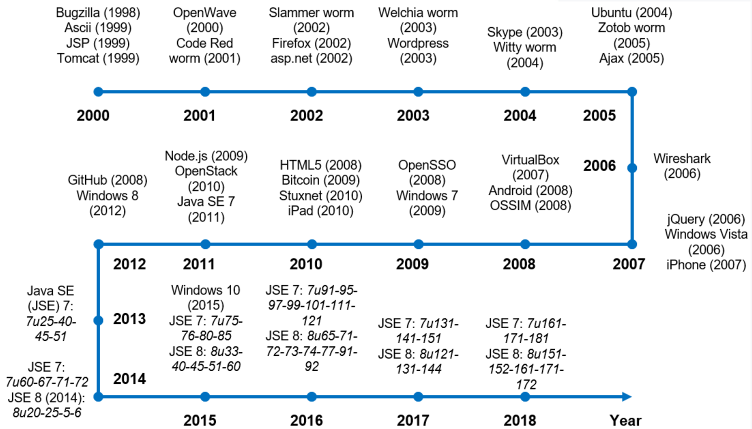
- Unending battle between attackers & defenders
- Evolving threat landscapes ➔ *Changing characteristics* of security issues over time

Causes

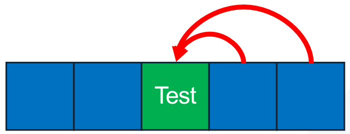
- New terms for emerging attacks, defenses, & issues
- Changing software features & implementation over time

Impacts

- Out-of-Vocabulary words ➔ Degrading performance
- Data leakage ➔ Unrealistic performance (up to ~5 times overfitting) using non-temporal evaluation technique



New threats/affected products in NVD over time



Non-temporal evaluation ➔ (Future) data leakage

THE UNIVERSITY of ADELAIDE

R. Croft et al., *Proceedings of the IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2023, 121-133.
R. Croft, et al., *Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2022, 338-348.
T. H. M. Le, et al., *Proceedings of the 16th International Conference on Mining Software Repositories*, 2019, 371-382.

Data (In)Accessibility

Info

- Security data not always shared
- Shared yet incomplete data ➔ Re-collected data may be different from original data

Causes

- Privacy concerns (e.g., commercial projects) or not?!
- Too large size for storage (artifact ID vs. artifact content)
- Data values can change over time (e.g., devs' experience)

Impacts

- Limited reproducibility of the existing results
- Limited generalisability of the ML models (e.g., open-source vs. closed-source data)

"We have anonymously uploaded the database to _https://www.dropbox.com/s/anonymised_link_ so the reviewers can access the raw data during the review process. We will release the data to the community together with the paper."

Click the link

THE UNIVERSITY of ADELAIDE

R. Croft, et al., *Empirical Software Engineering*, 2022, 27: 1-52.
T. H. M. Le, et al., *Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2021, 109-118.

Data (Re-)Use

Info

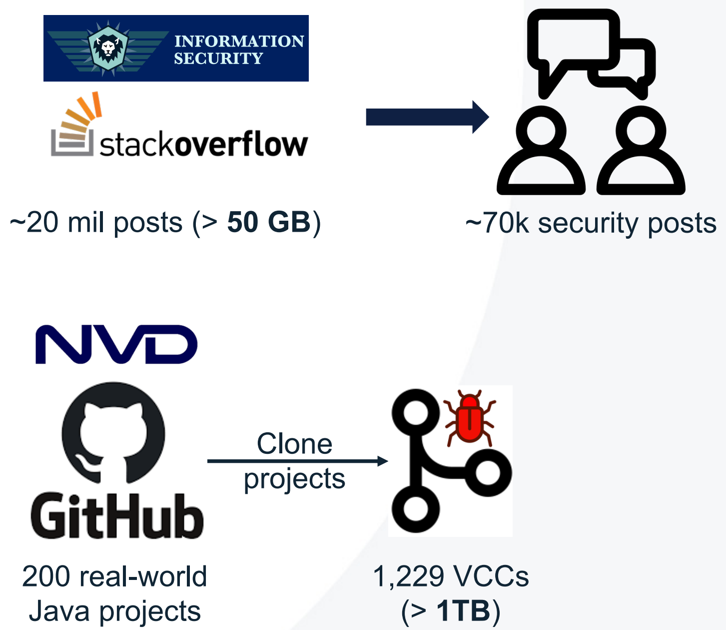
- Finding a needle (security issue) in a haystack (raw data)
- And ... haystack can be huge
- Reuse existing data > Update/collect new data

Causes

- Completely trusting/relying on existing datasets
- Unsuitable infrastructure to collect/store raw data

Impacts

- Making ML models become obsolete & less generalised
- Being unaware of the issues in the current data
  ➔ Error propagation in following studies



T. H. M. Le, et al., *Proceedings of the 36th International Conference on Automated Software Engineering (ASE)*, 2021, 717-729.
T. H. M. Le, et al., *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, 350-361.

THE UNIVERSITY of ADELAIDE

Data Maliciousness

Info

- Threat/security data is itself a threat (e.g., new vulns)
- Using/sharing threat data without precautions

Causes

- Simply an oversight / afterthought
- Private experiment vs. Public disclosure (Big difference!)
- Open science vs. Responsible science

Impacts

- Violating ethics of the community
- Shared security data maybe exploited by attackers
- Affecting maintainers & users of target systems

Researchers develop a SOTA ML-based vulnerability prediction model

↓

They identify *new vulnerabilities* using the model

↓

They don't report the vulnerabilities to project maintainers to fix

↓

They move on to submit & publish the paper

↓

The identified vulns later get exploited by attackers

An example of malicious/irresponsible data sharing

Data Quality for Data-Driven Software Security

Dataset	Accuracy	Uniqueness	Consistency	Completeness	Currentness
Big-Vul	0.543	0.830	0.999	0.824	0.761
Devign	0.800	0.899	0.991	0.944	0.811
D2A	0.286	0.021	0.531	0.981	0.844

- Prevalent noise in current real-world vulnerability datasets
- Significantly reduce SVP performance, e.g., data accuracy (30 – 80% ↓ in MCC)
- Automatic data cleaning: ↑ performance by ~20%, but still far from perfect

1. Data Quality for Software Vulnerability Datasets, ICSE '23 (CORE A*)
2. Data preparation for software vulnerability prediction: A systematic literature review, TSE '22 (A*)
3. Noisy label learning for security defects, MSR '22 (CORE A)
4. An investigation into inconsistency of software vulnerability severity across data sources, SANER '22 (CORE A)

Some Recommendations

THE UNIVERSITY *of* ADELAIDE

Recommendations for Dealing with Data Quality Issues

- Identification of Missing Vulnerability Data
  - Automatic labeling of silent fixes & latent vulnerabilities (beware of false positives)
- Consideration of Label Noise
  - Noisy Label Learning and/or Semi-Supervised Learning (small clean data & large unlabelled data)
- Consideration of Timeliness
  - Currently labeled data & more positive samples; Preserve data sequence for training
- Use of Data Visualization
  - Try to achieve better data understandability for non data scientists
- Creation and Use of Diverse Language Datasets
  - Bug seeding into semantically similar languages
- Use of Data Quality Assessment Criteria
  - Determine and use specific data quality assessment approaches
- Better Data Sharing and Governance
  - Provide exact details and processes of data preparation

Acknowledgements

- This talk is based on the research studies carried out by the CREST researchers, particularly by Roland Croft, Triet Le, Yongzheng Xie, Mehdi Kholoosi.
- Triet Le led the efforts for developing the content for this presentation
- Discussions in the SSI cluster of the CREST provided insights included in this presentation
- Our research partially funded by the Cybersecurity CRC
- We are grateful to the students, RAs and Open Source data communities

THE UNIVERSITY of ADELAIDE

make history.

THE UNIVERSITY of ADELAIDE

Contact: Ali Babar
ali.babar@adelaide.edu.au

CREST

CYBER SECURITY COOPERATIVE RESEARCH CENTRE